CLAIMS

What is claimed is:

1. A computer-implemented method for analysis of executable program code, the executable program including segments of code that correspond to callable functions in source code from which the executable code was generated, comprising:

reading from the executable program code pairs of entry points and endpoints, each pair including an entry point and an endpoint that are associated with a callable function in the source code and corresponding to a segment of the executable program code; and

generating analysis data for the functions identified by the pairs of entry points and end points.

2. The method of claim 1, further comprising scanning the executable program code for selected characteristics using the pairs of entry points and endpoints.

3. The method of claim 1, further comprising:

executing the program code;

detecting execution of the functions using the pairs of entry points and endpoints; and

recording selected execution characteristics of each executed function.

4. The method of claim 1, wherein the executable program code includes one or more dynamic load modules, the method further comprising:

reading entry points of initializer and deinitializer functions from dynamic load modules;

pairing the entry points of the initializer and deinitializer functions with endpoints of the initializer and deinitializer functions; and

generating analysis data for the initializer and de-initializer functions identified by the pairs of entry points and end points of the initializer and deinitializer functions.

5. The method of claim 4, wherein the executable program code includes a procedure lookup table (PLT) table associated with the one or more dynamic load modules, the method further comprising:

reading function entry points from the PLT;

pairing the entry points from the PLT with endpoints; and

generating analysis data for the PLT functions identified by the pairs of entry points and end points of the PLT functions.

6. The method of claim 4, further comprising scanning the executable program code for selected characteristics using the pairs of entry points and endpoints.

7. The method of claim 4, further comprising:

executing the program code;

detecting execution of the functions using the pairs of entry points and endpoints; and

recording selected execution characteristics of each executed function.

8. The method of claim 4, wherein the program code includes a symbol table identifying one or more function entry points, the method further comprising:

reading entry points of functions from the symbol table;

pairing the entry points from the symbol table with endpoints; and

generating analysis data for the symbol table functions identified by the pairs of entry points and end points of the symbol table functions.

9. The method of claim 1, wherein the program code includes a symbol table identifying one or more function entry points, the method further comprising:

reading entry points of functions from the symbol table;

pairing the entry points from the symbol table with endpoints; and

generating analysis data for the symbol table functions identified by the pairs of entry points and end points of the symbol table functions.

10. The method of claim 1, further comprising:

detecting function calls at runtime;

finding the entry point of a runtime-detected function call;

pairing an endpoint with the entry point of a runtime-detected function call; and

generating analysis data for functions identified by pairs of entry points and end points of the runtime-detected function calls.

11. The method of claim 10, further comprising:

detecting execution of stub functions at runtime; and

bypassing analysis of stub functions.

12. The method of claim 1, further comprising:

detecting execution of stub functions at runtime; and

bypassing analysis of stub functions.

13. The method of claim 10, wherein the executable program code includes one or more dynamic load modules, the method further comprising:

reading entry points of initializer and deinitializer functions from dynamic load modules;

pairing the entry points of the initializer and deinitializer functions with endpoints of the initializer and deinitializer functions; and

generating analysis data for the initializer and de-initializer functions identified by the pairs of entry points and end points of the initializer and deinitializer functions.

14. The method of claim 13, wherein the executable program code includes a procedure lookup table (PLT) table associated with the one or more dynamic load modules, the method further comprising:

reading function entry points from the PLT;

pairing the entry points from the PLT with endpoints; and

generating analysis data for the PLT functions identified by the pairs of entry points and end points of the PLT functions.

15. An apparatus for analysis of executable program code, the executable program including segments of code that correspond to callable functions in source code from which the executable code was generated, comprising:

means for reading from the executable program code pairs of entry points and endpoints, each pair including an entry point and an endpoint that are associated with a callable function in the source code and corresponding to a segment of the executable program code; and

means for generating analysis data for the functions identified by the pairs of entry points and end points.